Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні “Бази даних”

«Інформаційна система «Інтернет-магазин електроніки»»

Керівник, доц. каф ПІ Мазурова О.О.

Студент гр. ПІ-15-3 Крилов К.Ю.

Комісія: Мазурова О.О.

Зибіна К.В.

Черепанова Ю.Ю.

Харків 2016\_\_\_\_\_\_Харківський національний університет радіоелектроніки\_\_\_\_\_\_\_\_\_\_

**Кафедра** \_\_\_Програмної інженерії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Дисципліна**\_\_\_Базиданих\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Спеціальність**\_Програмнаінженерія\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Курс** \_\_\_2\_\_\_\_\_\_\_\_\_\_\_

**Група**\_\_\_ПІ-15-3\_\_\_\_\_\_\_\_\_**Семестр**\_\_\_\_4\_\_\_\_\_\_\_\_\_\_\_

**ЗАВДАННЯ**

**на курсову роботу студента**

\_\_\_\_\_Крилов Кирило Юрійович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Тема роботи:** \_\_\_\_«Інтернет-магазин електроніки»\_\_\_\_\_\_\_\_\_\_\_\_

**2. Строк здачі закінченої роботи** \_\_\_28.12.2016\_\_\_\_\_\_\_\_

**3. Вихідні дані для роботи:** методичні вказівки до виконання курсової роботи,\_вимоги до інформаційної системи, предметна область, що пов’язана з\_діяльністтю санаторія\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. Зміст розрахунково - пояснювальної записки:** вступ, аналіз предметної області; постановка задачі; проектування бази даних; опис програми; висновки; перелік посилань.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5. Перелік графічного матеріалу:** загальна схема концептуальної моделі, ER-діаграма, структура 1НФ, 2НФ, 3НФ, схема БД в 3НФ, UML-діаграми, копії екранів (“скріншоти”) прикладної програми, приклади звітів прикладної програми\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Дата видачі завдання** \_\_\_09.09.16 р.\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Назва етапів курсової роботи | Строк виконання етапів роботи | Примітки |
| 1 | Аналіз предметної області | 9.09.16 – 24.09.16 | Виконано |
| 2 | Постановка задачі | 20.09.16 – 30.09.16 | Виконано |
| 3 | Побудова ER-діаграми бази даних | 27.09.16 – 15.10.16 | Виконано |
| 4 | Оформлення розділів 1, 2 та 3.1, 3.2 пояснювальної записки | 15.10.16 - 27.10.16 | Виконано |
| 5 | Перша контрольна точка з курсового проекту | 24.10.16 – 28.10.16 | Виконано |
| 6 | Нормалізація бази даних | 20.10.16 - 10.11.16 | Виконано |
| 7 | Створення демо-версії програми | 20.10.16 – 20.11.16 | Виконано |
| 8 | Тестування програми, наповнення бази даних | 15.11.16 - 25.11.16 | Виконано |
| 9 | Друга контрольна точка з курсового проекту | 21.12.16– 02.12.16 | Виконано |
| 10 | Реалізація остаточної версії програми | 1.12.16-15.12.16 | Виконано |
| 11 | Оформлення інших розділів пояснювальної записки | 1.11.16 – 15.12.16 | Виконано |
| 12 | Захист курсового проекту (третя контрольна точка) | 12.12.16- 23.12.16 | Виконано |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Крилов К.Ю.

Керівник \_\_\_\_ доц. Мазурова О.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 р.

РЕФЕРАТ

Пояснювальна записка: 25с., 18 рис., 3 джерела.

Метою роботи є розробка інформаційної системи для iнтернет-магазину електронiки.

Метод розробки ­– концептуальне, ER- і UML- моделювання предметної області, об’єктно-орієнтований підхід до розробки програмного додатку.

Результати: спроектували базу даних і створили інформаційну систему для роботи з даними.

БАЗА ДАНИХ, ВІДНОШЕННЯ, СУБД, ІНФОРМАЦІЙНА СИСТЕМА, IНТЕРНЕТ-МАГАЗИН ЕЛЕКТРОНIКИ, PYTHON, SQL SERVER.

СОДЕРЖАНИЕ

Введение……………………………………………………………………. …….6

1 Анализ и концептуальное моделирование предметной области...……..…....7

1.1 Анализ предметной области……………………………………. …….. 7

1.2 Концептуальное моделирование……………………………………. 8

2 Постановка задачи………………………………………………………… 9

3 Проектирование базы данных………….…………………………………… 10

3.1 UML-моделирование……………………………………………………. 10

3.2 Построение ER-диаграммы……………………………………........... 11

3.3 Построение базы данных в третьей нормальной форме………………. 11

4 Описание программы…………………………………………………………16

4.1 Общие сведения……………………………………………………………..16

4.2 Вызов и загрузка…………………………………………………………….16

4.3 Назначение и логическая структура……………………………………….16

4.4 Описание физической модели базы данных …………………………….. 16

4.5 Описание программной реализации……………………………………….17

Выводы…………………………………………………………………….…….26

Перечень ссылок…….………………………......................................................27

ВВЕДЕНИЕ

В данной курсовой работе будет рассмотрена информационная система «Интернет-магазин электроники».

Данная система предназначена для просмотра и покупки пользователями электронных приспособлений. Система позволит продавцам вести учет товаров, собирать статистику о них. Покупатели смогут просмотреть всю необходимую информацию о товарах.

Популярность интернет-магазинов постоянно растет. Оценив удобства и преимущества покупок в режиме онлайн, многие люди, не выходя из дома, приобретают самые различные товары. Сэкономив деньги и время, в любое время можно спокойно изучить весь имеющийся ассортимент и спокойно заказать необходимый товар.

Онлайн-магазины работают круглосуточно и без выходных, что является большим преимуществом. Нет необходимости подстраиваться под график магазина - можно осуществлять покупки в любое удобное время суток. Товар заказывается сразу же на сайте, нужно лишь заполнить необходимую форму.

Удобная схема работы интернет-магазинов выгодна как их создателям, так и покупателям, поэтому количество покупок, проходящих через онлайн-сервис, неизменно растет, радуя качеством и ценой даже самых требовательных покупателей.

Разрабатываемая информационная система «Интернет-магазин электроники» предназначена для торговли электроникой в интернете. Учитывая специфику данной области, основными целями являются реализация удобного пользовательского интерфейса и возможности учета товарооборота.

В данной курсовой работе использована такая СУБД как MySQL Server, поскольку она является наиболее оптимальным решением для поставленного вопроса. Информационная система реализована с помощью Django (Python web framework). Среда разработки - PyCharm.

1 АНАЛИЗ И КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

В данной курсовой работе будет рассмотрена информационная система «Интернет-магазин электроники».

Цель данной системы – обеспечить комфортное исследование ассортимента товаров для дальнейшего их приобретения. Также она должна предоставлять возможности по изучению актуальности тех или иных товаров для администрации магазина.

Итак, в предметной области образуются следующие сущности:

а) категория и подкатегория продуктов;

б) продукт cо всеми атрибутами и изображениями;

в) пользователь сайта. Пользователь может быть обычным покупателем или иметь привилегии администратора, что позволит ему добавлять товары в базу данных, редактировать ее таблицы и просматривать статистики;

г) лайк, поставленный пользователем какому-либо продукту. В результате этот продукт появляется в списке избранных продуктов пользователя;

д) оценка пользователем продукта. На ее основе вычисляется рейтинг продукта, от которого зависит его приоритет;

е) корзина пользователя, в которую пользователь может добавлять продукты с целью дальнейшей покупки;

ж) заказ, представляющий набор продуктов, которые теперь следует доставить заказчику по указанным данным.

Информационные потребности пользователей данной информационной системы следующие:

Для покупателей:

а) поиск товаров по названию, типу, атрибутам, интервалам числовых атрибутов;

б) сортировка товаров по цене или рейтингу;

в) добавление товара в корзину;

г) добавление товара в избранное;

д) оценка товара;

е) фильтрация товаров по любому атрибуту;

ж) поиск товара по наименованию.

Для продавца:

а) добавление новых товаров в базу;

б) редактирование информации в базе данных;

в) получение отчетов по продажам за период: в стоимостном и количественном выражении в разрезе категорий товаров;

г) просмотр статистики о стоимости проданных товаров за последние 12 месяцев;

д) просмотр статистики о суммарной цене проданных товаров каждого производителя для каждой категории товаров за последний месяц.

Описание основных объектов ПО и их атрибутов:

а) Товар: Id товара, цена товара, название товара, описание товара, id подкатегории, id титульного изображения;

б) Заказ: Id заказа, адрес доставки, id состояния заказа, контактный номер, имя получателя;

в) Покупатель: Id покупателя, пароль, email, дата регистрации;

г) Корзина: Id корзины, Id покупателя, Id заказа;

д) Покупка: Id товара, Id корзины, кол-во товара

Алгоритмические зависимости:

а) расчет рейтинга товара как среднее по всем его оценкам;

б) рекомендованные товары исходя из недавно посещенных страниц товаров и их рейтингов;

Для выполнения поставленных задач и целей будет создана информационная система «Интернет-магазина электроники» средствами фреймворка Django. Приложение будет разработано в среде PyCharm с использованием СУБД MySQL server.

2 ПОСТАНОВКА ЗАДАЧИ

Необходимо реализовать процесс взаимодействия покупателя с информационной системой, а также процесс взаимодействия администратора и системы.

Набор пользовательских возможностей у покупателя будет ограничиваться просмотром, оцениванием товаров и добавлением их в избранное/корзину/заказ. Для этого также необходимо реализовать систему регистрации.

Набор возможностей администратора будет состоять из редактирования содержимого магазина, просмотра заказов, формирования отчетов и статистических данных.

Система должна выполнять следующие задачи:

а) отображение товаров по категориям;

б) выведение информации о конкретном товаре;

в) поиск товаров по названию;

г) фильтрация товаров по наличию тех или иных атрибутов, по интервалам атрибутов числового типа;

д) сортировка товаров по цене, рейтингу;

з) вывод всех когда-либо совершенных покупателем заказов;

и) возможность добавления новых товаров в базу данных;

й) возможность редактирования информации о товарах;

к) оценивание товаров пользователем;

л) добавление пользователем товаров в избранное;

м) формирование диаграмм по объемам продаж производителей товаров в общем объеме продаж в разрезе категорий товаров;

н) формирование графика продаж в стоимостном выражении за последние 12 месяцев по месяцам;

о) формирование отчетов по продажам за последние 30 дней в стоимостном и количественном выражении в разрезе категории товаров.

3 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

3.1 Построение Use Case диаграммы

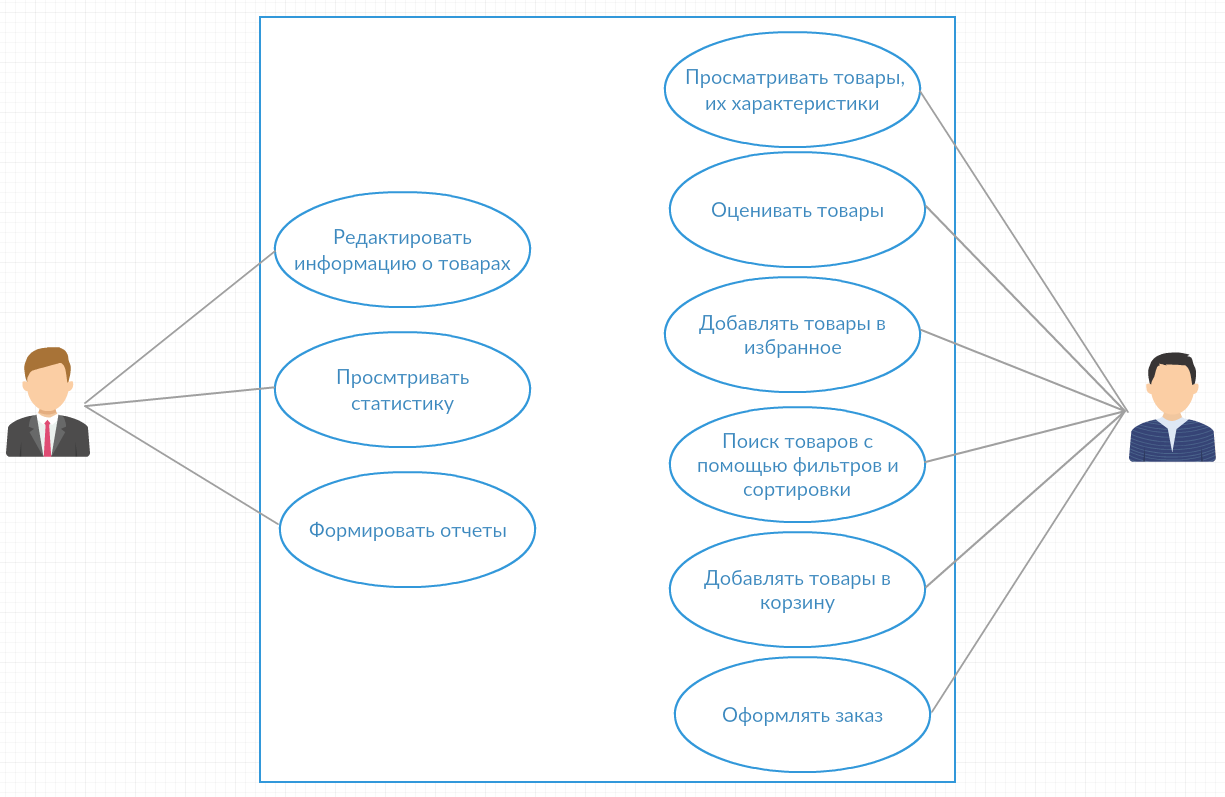
В Use Case диаграмме, на рисунке 3.1, отображается взаимодействие клиента, продавца и информационной системы. При этом клиент не имеет доступа к функциям, изменяющим содержимое магазина. 

Рисунок 3.1 – Use Case диаграмма

3.2 Построение ER-диаграммы

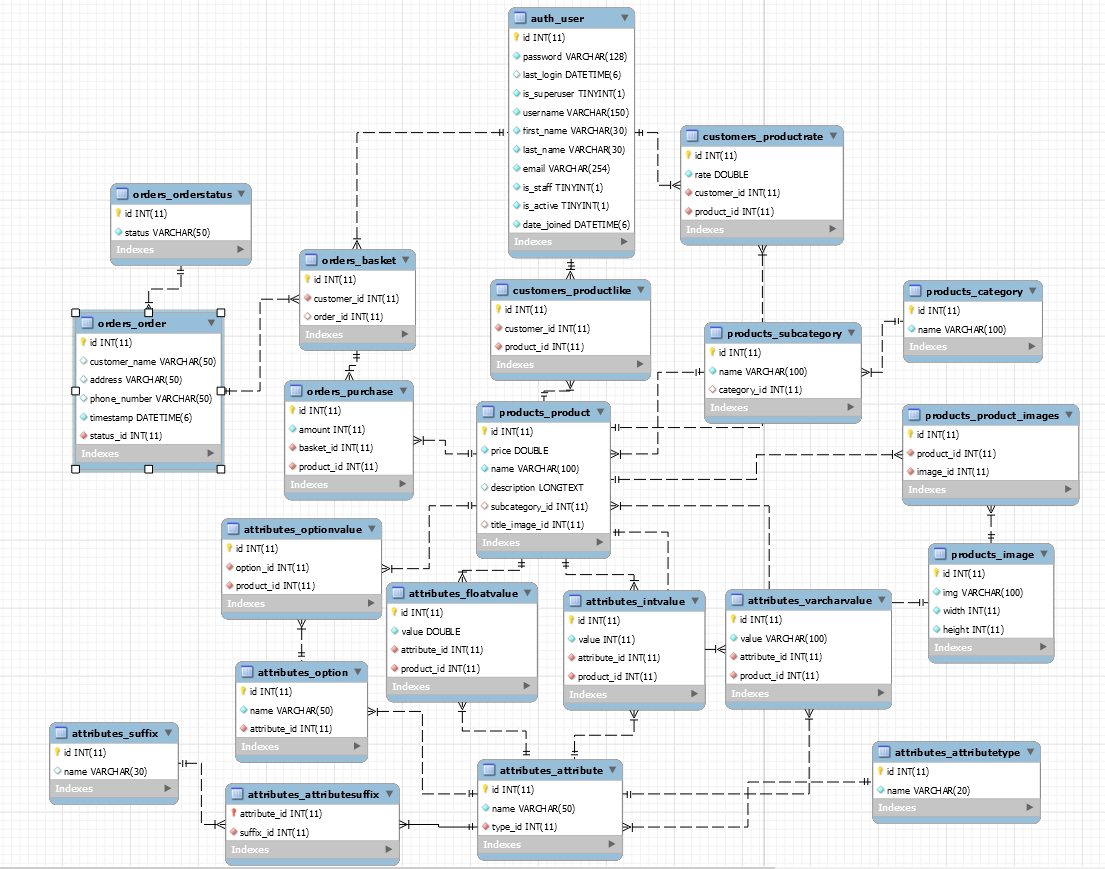


Рисунок 3.2 – ER-диаграмма

3.3 Построение схемы реляционной базы данных в третьей нормальной форме:

Нормализация – разделение таблицы на две или больше с целью исключения избытка информации и избегания аномалий баз данных.

1НФ представляет собой отношение. Для него характерны атомарность значений на пересечении столбцов и строк, кортежи и атрибуты должны быть неупорядочены и кортежи не дублируются.

Для 2НФ характерны особенности 1НФ и полная функциональная зависимость неключевых атрибутов от первичного ключа отношения.

3НФ – состояние отношения, которое находится во 2НФ и в котором между атрибутами нет транзитивных зависимостей.

Переход к следующей нормальной форме возможен только тогда, когда удовлетворяются все условия предыдущей формы.

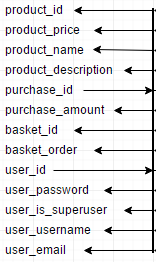
Для предметной области построим универсальное отношение, которое включает описание ранее выделенных объектов и их свойств (см. рис. 3.3).   


Рисунок 3.3 – Универсальное отношение

Данное универсальное отношение удовлетворяет требования 1НФ, то есть является отношением реляционных баз данных, а именно: кортежи не упорядочены, атрибуты не упорядочены, информация кортежей не дублируется, значения атрибутов атомарные. Это универсальное отношение не является 2НФ, так как атрибуты «product\_id», «product\_price», «product\_name», «product\_description», «purchase\_amount», «basket\_id», «basket\_order», «user\_passsword», «user\_is\_superuser», «user\_username», «user\_email» не полностью зависят от первичного ключа. Продолжим процедуру нормализации.

2НФ будет состоять из двух таблиц.

Выделим таблицу Т1, рисунок 3.4, и таблицу Т2, рисунок 3.5.

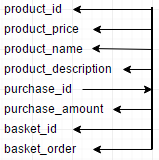


Рисунок 3.4 – Таблица Т1

Таблица Т1 находится в 2НФ, так как все атрибуты имеют полную функциональную зависимость от первичного ключа «purchase\_id». Также присутствуют транзитивные зависимости, что значит таблица не в 3 НФ.

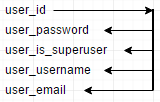


Рисунок 3.5 – Таблица Т2

Таблица Т2 находится в 2НФ и 3НФ, т.к. не имеет транзитивных зависимостей.

Для привидения Т1 к 3НФ разделим её на Т3, рисунок 3.6 и Т4, рисунок 3.7.

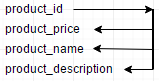


Рисунок 3.6 – Таблица Т3

Таблица Т3 находится в 3НФ, так как она находится в 2НФ и в ней отсутствуют транзитивные зависимости.

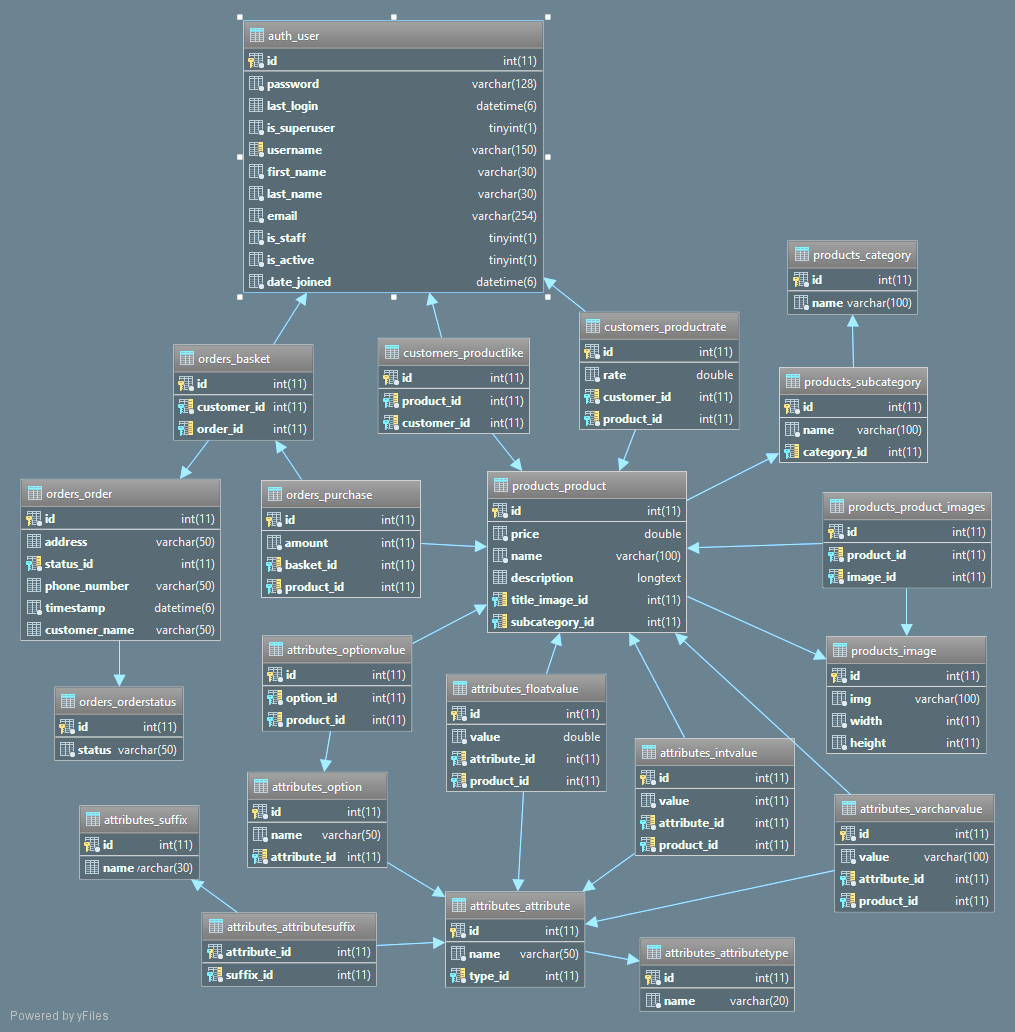
D:\Workspace\cloud\store\Capture13.PNG

Рисунок 3.7 – Таблица Т4

Таблица Т4 находится в 3НФ, так как она находится в 2 НФ и в ней отсутствуют транзитивные зависимости.

Данные таблицы далеки от представленных в ER-диаграмме, так как хоть они и в 3НФ, но после построения связей набор атрибутов изменится, но это не отменит того, что они будут в 3НФ. Это можно увидеть на конечной схеме базы данных (в 3НФ), рисунок 3.8.

Рисунок 3.8 – Схема базы данных



4 ОПИСАНИЕ ПРОГРАММЫ

4.1 Общие сведения

Разрабатываемый программный продукт создавался в интегрированной среде разработки для языка программирования Python PyCharm. В качестве СУБД была выбрана MySQL Server. Ее основные преимуществами, безусловно, являются простота в использовании (упрощенное создание таблиц, отстутствие сложных операций управления базой данных, и т.д.), а также то, что она является одной из самых надежных платформ базы данных. Эта СУБД очень популярна, а значит она хорошо развита и содержит минимум ошибок.

4.2 Вызов и загрузка

Для установки сервера необходим django 1.7 и python 3.5.2. Сервер запускается командой python manage.py runserver и доступен через порт 8000 по умолчанию.

4.3 Описание физической модели БД

Для того, чтобы реализованная система фильтров функционировала, значения атрибутов хранятся в четырех таблицах: таблице целых значений (intvalue), дробных (floatvalue), строчных (varcharvalue) и таких, для которых список возможных значений относительно постоянен (optionvalue).

Для числовых значений приходится хранить т.н. «суффикс» - единицу измерения, приписываемую к числу (например, Mpx или Gb). Для этого существует таблица suffix и таблица attribute\_suffix, соединяющая ее с таблицой атрибутов, т.к. одна и та же единица измерения может быть у нескольких атрибутов.

Таблицы purchase устанавливает связь many-to-many между таблицей продуков и таблицей корзин пользователей.

Поле order\_id таблицы basket принимает не NULL значение, если она представляет совершенный заказ, и NULL, если она, собственно, представляет корзину.

Таблица order\_status содержит возможные состояния заказа (доставлен – не доставлен).

Таблицы product\_rate и product\_like устанавливают отношение many-to-many между пользователем и товаром и представляют оценку товара и лайк соответственно.

4.4 Описание программной реализации

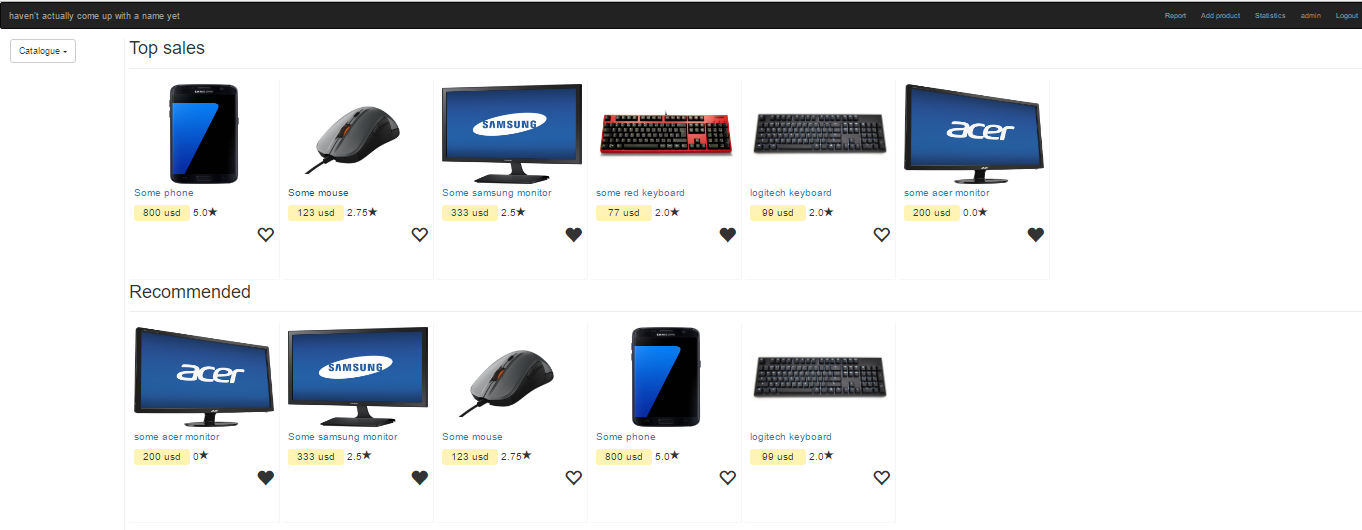


Рисунок 4.2 – Главная страница

На рисунке представлена главная страница сайта. На ней представлены топ товаров по рейтингу, а также рекомендации, основанные на недавней истории просмотров. В верхнем правом углу расположены кнопки логина и регистрации или выхода и личного кабинета (если вход произведен). На левой панели находится кнопка с выпадающим списком всем категорий и их подкатегорий (рис. 4.3). Любой продукт можно добавить или удалить из избранного кликом по пиктограмме под его изображением.

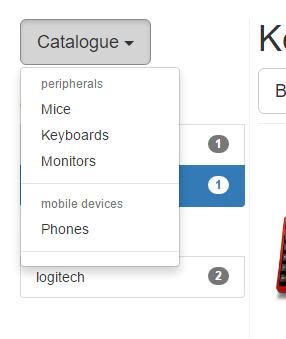
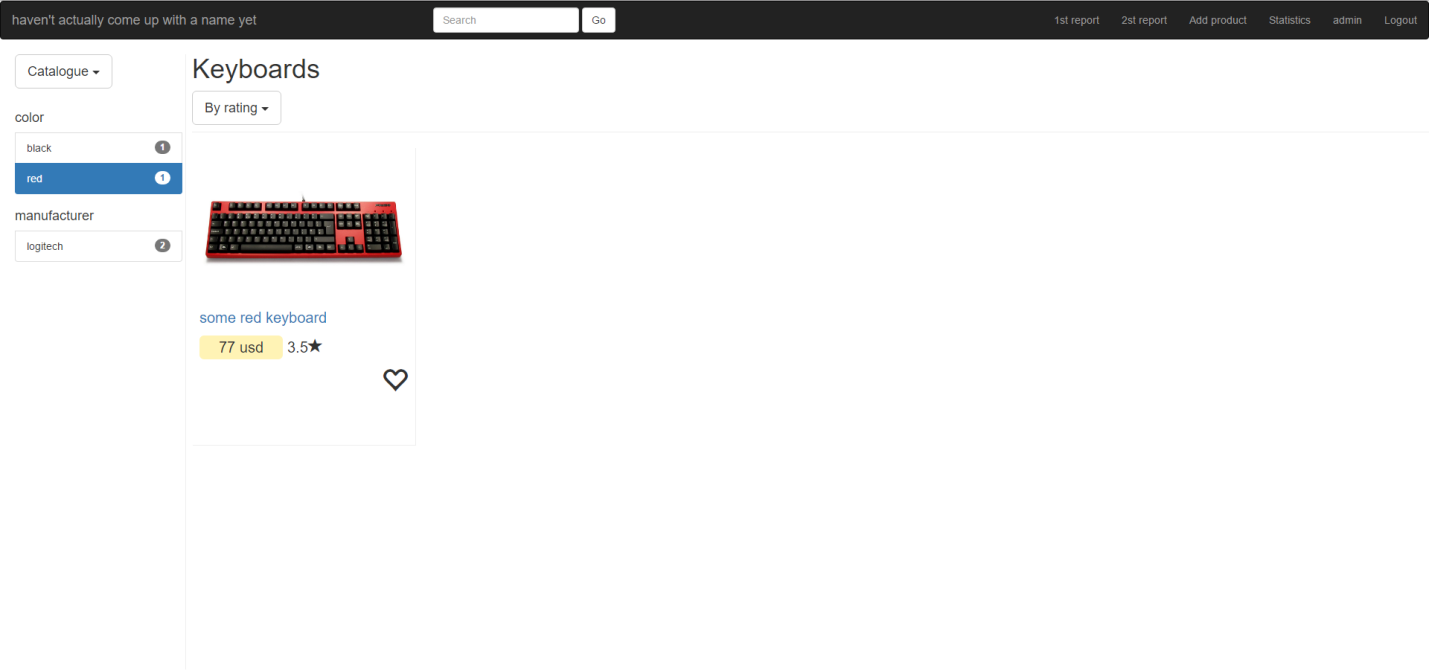


Рисунок 4.3 – Меню категорий

Рисунок 4.4 – Вид подкатегории

На рисунке представлена страница, на которую попадает пользователь при выборе подкатегории. На ней доступен фильтр по названию вверху страницы и фильтры по атрибутам слева. На каждом атрибуте написано число – количество продуктов, имеющих данный атрибут. Клик по атрибуту включает или исключает все товары, имеющие его. Под названием подкатегории расположена кнопка выбора сортировки (рис 4.5)

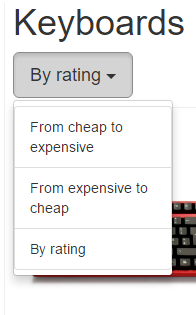


Рисунок 4.5 – Сортировка

При клике по товару с любой страницы сайта, пользователь переходит на страницу выбранного товара (рис. 4.6). На ней отображена вся доступная информация: название товара, цена, рейтинг, описание, атрибуты, изображения. На ней пользователь может оценить товар или добавить его в корзину.

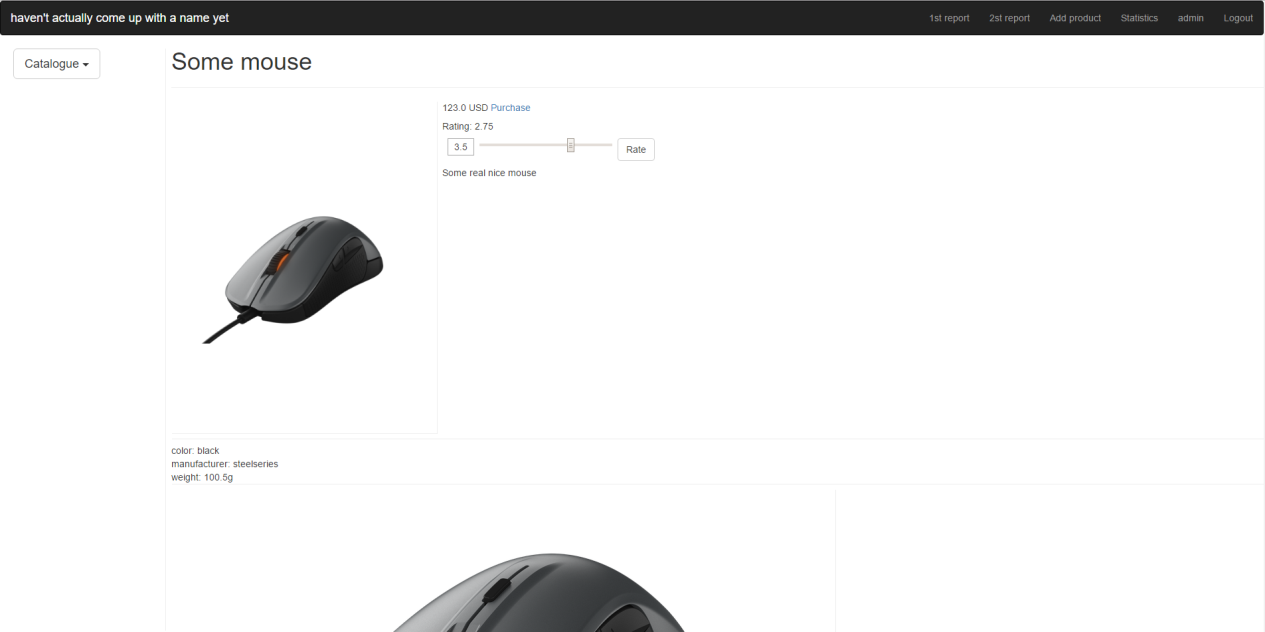


Рисунок 4.6 – Информация о конкретном товаре

Клик по своему имени в верхнем правом углу ведет пользователя в его личный кабинет, где он сразу видит корзину (рис. 4.7). На левой панели расположена форма оформления заказа, где пользователь может ввести данные, необходимые для доставки товара.

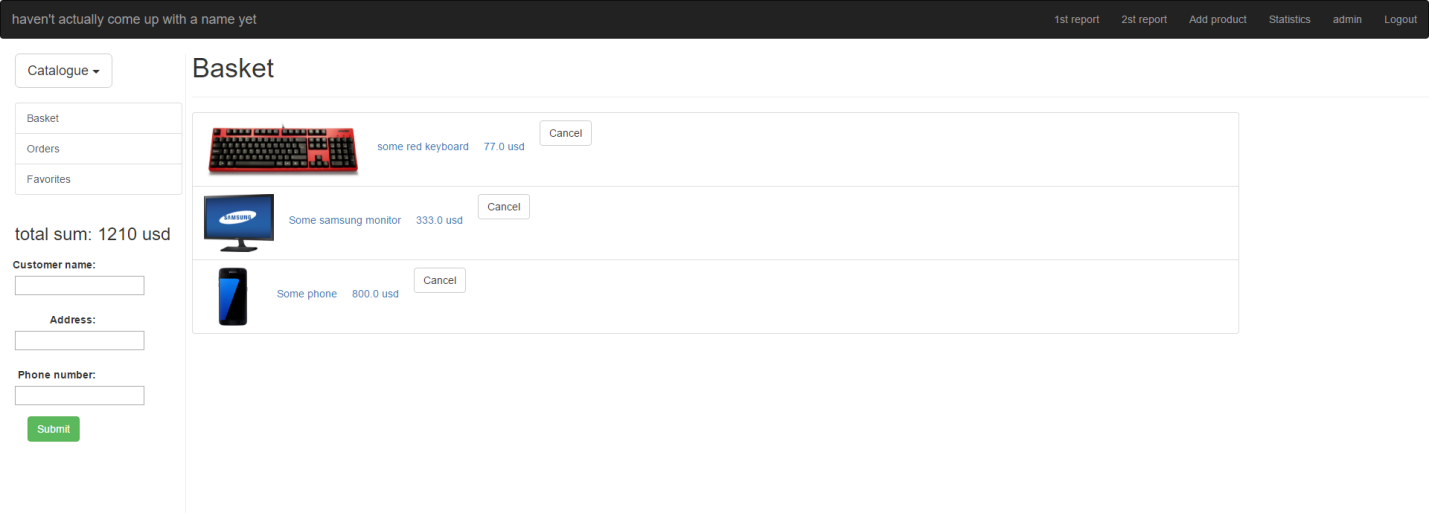


Рисунок 4.7 – Корзина покупок

В левой части экрана также есть панель навигации по кабинету: ссылки на корзину, заказы и избранное. На странице заказов (рис. 4.8) представлен список когда-либо оформленных заказов и информация о них. Еще не доставленные заказы можно отменить.

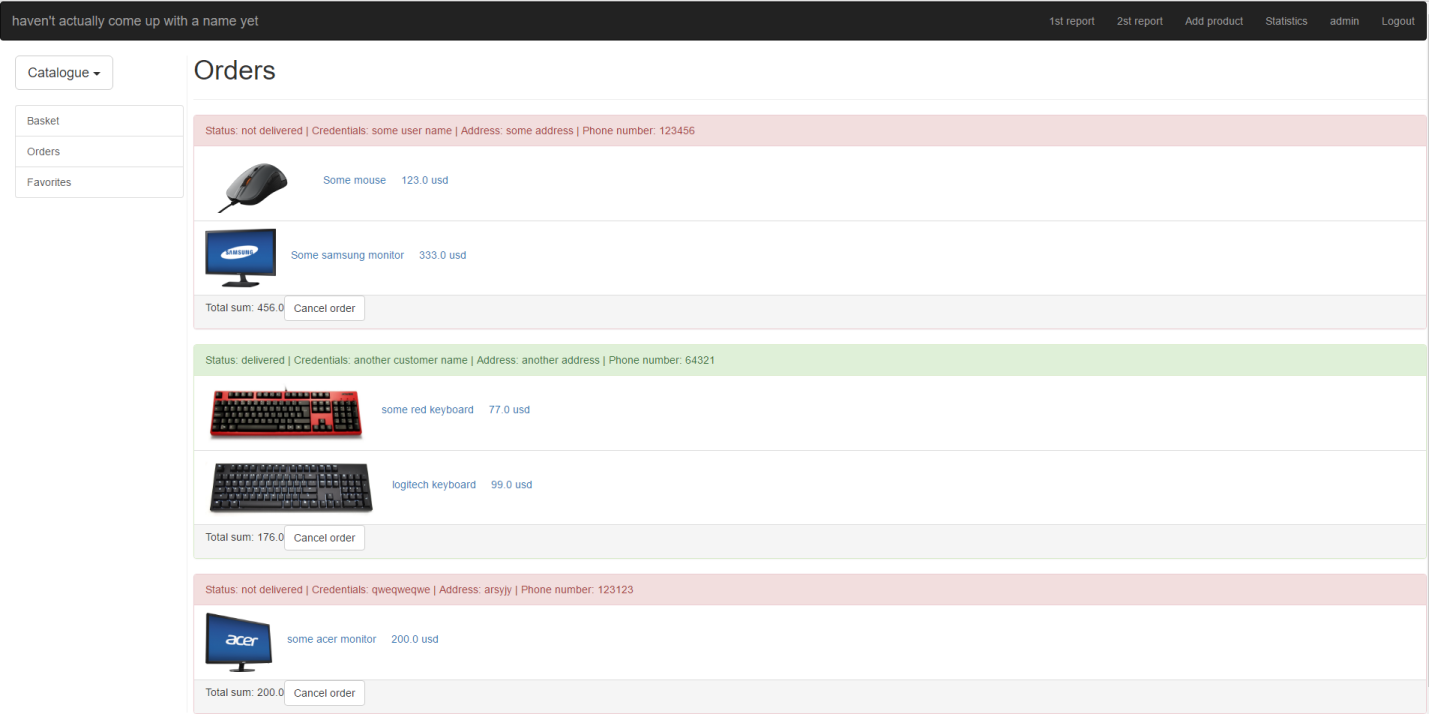


Рисунок 4.8 – Список заказов

Если пользователь зашел как администратор, в верхнем правом углу у него будут опции добавления нового товара, просмотра статистик (рис. 4.9) и отчетов (рис. 4.10, 4.11).

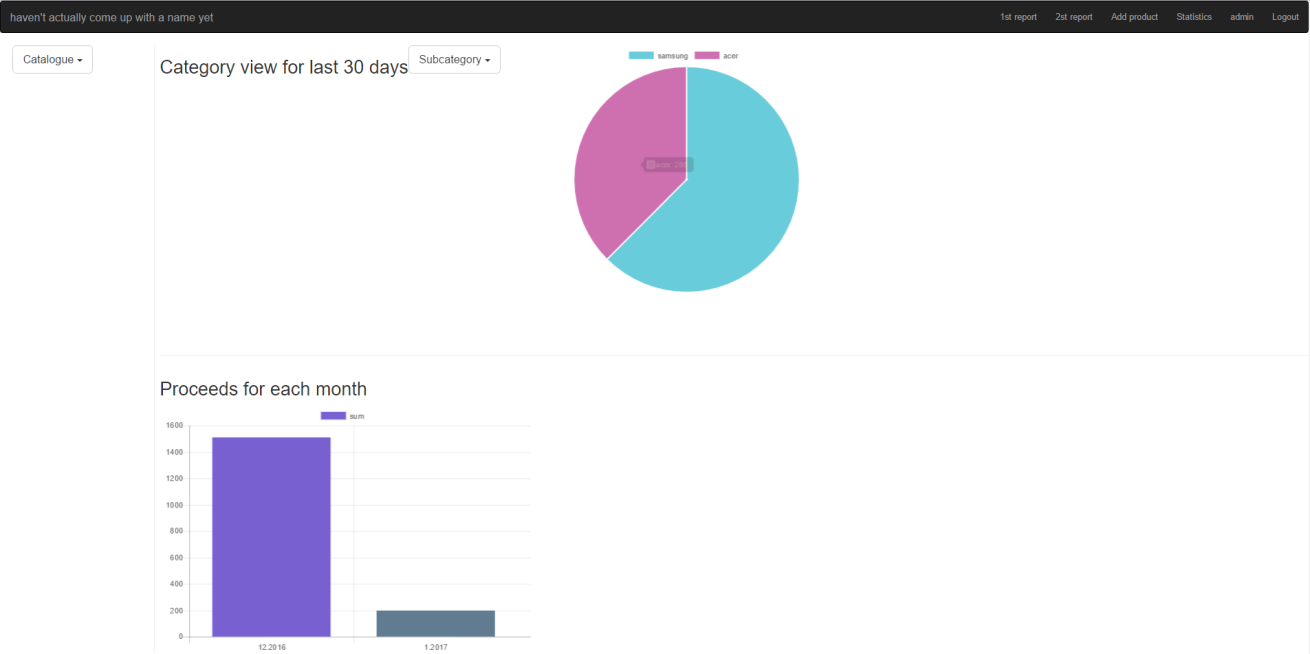


Рисунок 4.9 – Статистики

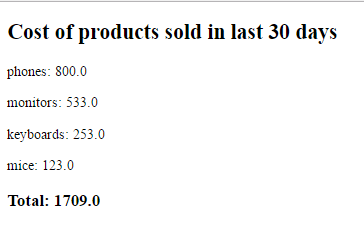
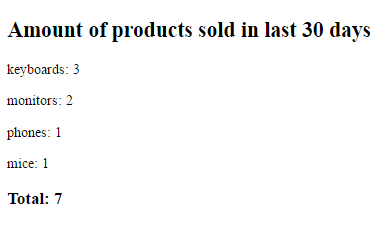
  
Рисунок 4.10 – Первый отчет  


Рисунок 4.11 – Второй отчет

Далее приведены примеры запросов, использованных для получения информации.

Фильтр товаров подкатегории по атрибутам и названию:

SELECT p.id, p.name, p.price, p.rating, COUNT(\*) AS count

FROM (SELECT p.id, p.name, p.price, p.rating, COUNT(\*) AS count

FROM (SELECT p.\*, if(pl.id IS NOT NULL, 1, 0) AS liked, if(pr.rating IS NULL, 0, pr.rating) AS rating

FROM products\_product p LEFT JOIN

(SELECT id, product\_id FROM customers\_productlike WHERE customer\_id = 1) AS pl ON p.id = pl.product\_id

JOIN products\_subcategory sc ON p.subcategory\_id = sc.id

LEFT JOIN (SELECT product\_id, AVG(rate) AS rating FROM customers\_productrate GROUP BY product\_id) AS pr

ON p.id=pr.product\_id

WHERE (sc.id = 4) AND (p.name LIKE %som%)) as p

LEFT JOIN attributes\_optionvalue ov ON p.id = ov.product\_id

LEFT JOIN attributes\_option o ON ov.option\_id = o.id

LEFT JOIN attributes\_attribute a ON o.attribute\_id = a.id

WHERE

(LOWER(a.name) = color AND LOWER(o.name) = black) OR (LOWER(a.name) = manufacturer AND LOWER(o.name) = acer) GROUP BY p.id, p.name, p.price, p.rating HAVING count = 2) as p

LEFT JOIN attributes\_floatvalue fv ON p.id = fv.product\_id

LEFT JOIN attributes\_attribute a ON fv.attribute\_id = a.id

WHERE

(LOWER(a.name) = diagonal AND (fv.value BETWEEN 20 AND 30)) GROUP BY p.id, p.name, p.price, p.rating HAVING count = 1 ORDER BY p.price DESC;

Получение атрибутов конкретного товара:

SELECT ov.id AS id, a.name AS name, o.name AS val

FROM attributes\_optionvalue ov

JOIN attributes\_option o ON ov.option\_id = o.id

JOIN attributes\_attribute a ON o.attribute\_id = a.id

WHERE ov.product\_id = 7

UNION

SELECT iv.id AS id, a.name AS name, CONCAT(iv.value, IF(s.name IS NOT NULL, s.name, '')) AS val

FROM attributes\_intvalue iv

JOIN attributes\_attribute a ON iv.attribute\_id = a.id

LEFT JOIN attributes\_attributesuffix atsuf ON atsuf.attribute\_id = a.id

LEFT JOIN attributes\_suffix s ON atsuf.suffix\_id = s.id

WHERE iv.product\_id = 7

UNION

SELECT fv.id AS id, a.name AS name, CONCAT(fv.value, IF(s.name IS NOT NULL, s.name, '')) AS val

FROM attributes\_floatvalue fv

JOIN attributes\_attribute a ON fv.attribute\_id = a.id

LEFT JOIN attributes\_attributesuffix atsuf ON atsuf.attribute\_id = a.id

LEFT JOIN attributes\_suffix s ON atsuf.suffix\_id = s.id

WHERE fv.product\_id = 7

UNION

SELECT vcv.id AS id, a.name AS name, vcv.value AS val

FROM attributes\_varcharvalue vcv;

JOIN attributes\_attribute a ON vcv.attribute\_id = a.id

WHERE vcv.product\_id = 7

Запрос, получающий рекомендованные товары:

SELECT p.id, p.price, p.name, p.description, p.subcategory\_id, liked, if(AVG(pr.rate) IS NULL, 0, AVG(pr.rate))\*p.hits AS relevance FROM

(SELECT p.\*, if(pl.id IS NOT NULL, 1, 0) AS liked, SUM(h.hits) AS hits FROM products\_product p

LEFT JOIN (SELECT id, product\_id FROM customers\_productlike WHERE customer\_id = 1) AS pl

ON p.id = pl.product\_id

JOIN attributes\_optionvalue ov ON p.id = ov.product\_id

JOIN ((SELECT id, 5 AS hits FROM attributes\_option WHERE id = 6) UNION (SELECT id, 2 AS hits FROM attributes\_option WHERE id = 3) UNION (SELECT id, 8 AS hits FROM attributes\_option WHERE id = 1)) AS h ON h.id = ov.option\_id

GROUP BY p.id, p.price, p.name, p.description, p.subcategory\_id) AS p

LEFT JOIN

customers\_productrate pr ON p.id=pr.product\_id

GROUP BY p.id, p.price, p.name, p.description, p.subcategory\_id

ORDER BY relevance DESC;

Запрос, получающий статистику по продажам каждого производителя по данной категории за последние 30 дней:

SELECT opt.id, opt.name AS manufacturer, SUM(pr.price\*pur.amount) AS sum FROM products\_product AS pr

JOIN orders\_purchase AS pur ON pr.id = pur.product\_id

JOIN orders\_basket AS b ON pur.basket\_id = b.id

JOIN orders\_order AS ord ON b.order\_id = ord.id

JOIN products\_subcategory AS sc ON pr.subcategory\_id = sc.id

JOIN attributes\_optionvalue AS ov ON pr.id = ov.product\_id

JOIN attributes\_option AS opt ON ov.option\_id = opt.id

JOIN attributes\_attribute AS a ON opt.attribute\_id = a.id

WHERE b.order\_id IS NOT NULL

AND a.name = 'manufacturer'

AND sc.name = 'monitors'

AND ord.timestamp BETWEEN (CURDATE() - INTERVAL 30 DAY) AND (CURDATE() + INTERVAL 1 DAY)

GROUP BY opt.id, opt.name;

ВЫВОДЫ

В данной курсовой работе была разработана информационная система «Интернет-магазин электроники». Она была создана в форме веб-приложения с серверной частью, написанной на Python-фреймворке Django.

Данная система является упрощенным прототипом реального интернет-магазина и не может использоваться как готовый продукт, поскольку была создана в учебных целях.

Работа над информационной системой требовала детального разбора будущей базы данных, в ходе которого были изучены различные этапы оптимизации базы данных и проекта в целом. Среди них – ER-диаграмма, нормализация, концептуальный анализ, и т.д.

Все это в совокупности позволило получить качественную и работоспособную информационную систему, что удовлетворяет все требования, которые были поставлены. Наиболее ценным в этом проекте являются приобретенные навыки работы с различными СУБД и закрепление знаний в направлении web-разработки. Кроме того, использование SQL-запросов, и т.д., несомненно, способствуют разностороннему и гармоничному развитию студента как программиста.

ПЕРЕЧЕНЬ ССЫЛОК

1. Дюбуа П. MySQL Сборник рецептов[Текст] / П. Дюбуа. – Санкт-Петербург: Символ-Плюс. – 1056 с. – (Вне серии).
2. Тахагхогхи С. Руководство по MySQL[Текст] / С. Тахагхогхи, Х. Вильямс. – Москва: Русская редакция, 2007. – 544 с. – (Бестселлеры O`Reilly).
3. Гольцман В. И. MySQL 5.0. Библиотека программиста[Текст] / В. И. Гольцман., 2010. – 253 с. – (Библиотека программиста).